DB2 PROJECT 1 DESIGN REPORT

RIGOROUS 2PL & WOUND WAIT METHOD

Nikitha Chennamaneni -1001745322

Project Description

The implementation of this will make use of HashMap class of JAVA for storing and processing the transactions and the locks applied on the data items.

Language used:

Java

Data Structures:

The implementation of the simulation will make use of HashMap class of JAVA for storing and processing the transactions and the locks applied on the data items.

The following are the table structures that will be used to capture the transaction and locking information:

Transaction table: It contains details such as transaction id, transaction time stamp, transaction state, a list of data items held in FIFO order.

The data structure used is Hash table**.**

Key – Transaction Id

Value – Transaction object – This contains the details about the transaction such as:

- String transactionState : This is used to store the status of the transaction like 'Active', 'Blocked', 'Aborted', 'Committed'
- int transactionId : To store the Transaction ID of aparticular transaction
- int timeStamp: To store the Time stamp
- List itemsHeld = new ArrayList() This is the store the items held by the transaction.
- Queue of Operation objects waitingOperations using linked list;

Lock table: It contains details such as data item name, type of lock, list of transactions accessing the item and a list of waiting transactions.

The data structure used is Hash table**.**

Key – Data item

Value – Lock object - This contains the details such as:

- String itemName: This stores the Data Item
- String lockState: This is the store operation like ‘read’ or ‘write
- PriorityQueue<String> readLockTransId to hold the read locked transaction ids

  String writeLockTransId to hold the write locked transaction id.
- List<Integer waitingTID = new ArrayList<Integer>(): To store Waiting Transation IDs >

Implementation:

Step 1. Read from file

The program starts from Main method and the input file is read line by line. As we read a line in the in the input file we will store Transaction Id(TID), operation(begin(b),read(r),write(w)..), data item.

The program consists of a switch case which calls methods for begin transaction - ‘b’, read operation - ‘r’, write operation - ‘w’ and end transaction - ‘e’.

Pseudocode:

```
Main ()
{
        readInput()
        {
                1.read input file line by line

                2.if  line starts with “b” then
                        call beginTransaction(Transaction id, Time stamp, “ACTIVE”)

                3.if  line starts with “r” then
                        call readTransaction (Transaction id, data item)

                4.if  line starts with “w” then
                        call writeTransaction (Transaction id, data item)

                5.if line starts with “e” then
                        call end or abortTransaction (Transaction id, “COMMIT”)
        }
}
```

Step 2. Begin Transaction

Class TransactionDetails uses HashMap- TransactionTable is used to store the following information: Transaction id, transaction timestamp, transaction state and List of items locked (Lists of Items held).

Status will be 'Active' for all new transactions. Timestamp will be incremented whenever a new transaction is read from input file. Initially set to 1, store the record in hashmap and display it.

Pseudocode:

beginTransaction (Transaction id)

{

1. Create a entry in the transaction table(HashMap) with key equals Transaction id and value is transaction object which stores the Timestamp and Transaction State and List of Items held.
2. Update the variables of the transaction object i.e, state is updated as "Active" and Timestamp is incremented by 1
3. Create Arraylists for storing list of data items held and list of waiting operations respectively.
4. The list of data items held is stored in HashSet and the list of and waiting operations are stored in the queue

}

Step 3.Read transaction function:

This Read function is called when a read operation - 'r' is encountered from the input file. The input to this fuction is the corresponding Transaction id and data item.

If the input file reads 'r' then

read function is called and read_lock on item is requested

Pseudocode:

readTransaction(TransactionId, dataItem)

{

If trans_State is equal to "Active"

If lock table contains the ItemName

Transaction in the lock table is updated to read-locked

If the item is not locked

An entry is created in the lock datastructure with, the lock state is set to read and the transaction id and itemName are read and displayed

End If;

Else if the item is already locked by a read operation/read-write conflict

readReadfuction is called for "Read Operation" and writeReadfuction is called for "Read-Write Conflict" by using wound wait prevention protocol respectively.

End Else;

End If;

Else

Create new lock with itemName and lock State, add the transaction id and store it in the lock table

End Else;

End If;

Else

if transaction is blocked

If lock table doesnot contains the ItemName

The lock state is set to write and the transaction id and itemName are read and displayedtransaction table. These operations will be executed once operation is resumed

End if

End If;

Else (Aborted)

Unlock any items that are locked and restart again with the same timestamp;

End Else;

End Else

}

Step 4. Write transaction function

If the input file reads ‘w’ then

‘write’ function is called then a write_lock on item is requested

Pseudocode:

writeTransaction(transactionId, dataItem){

If transactionState is equal to "Active"

If lock table contains the ItemName

If the lockstate of the item is empty

Lock state of the item is set to write and the item is added to the list of items held by the transaction

Endif

Else

if the item is already read locked/read write conflict

readWrite function is called to resolve the conflictby using wound wait prevention protocol.

End if

Else if the item is already read locked/read write conflict

writeWritefunction is called to resolve the conflictby using wound wait prevention protocol.

End else

End else

End if

Else

Create new lock with itemName and lock State, add the transaction id and store it in the lock table

End else

End if

Else

If the transaction state is blocked

if lock table doesnot contains the ItemName

Create an entry in the lck table with item name, lock state,transactio id as null.

End if

Transaction state is changed to 'blocked' and addoperations to thepriority queue in order that arewaiting to be executed in the transaction table. Theseoperations will be executed once operation isresumed

End if

Else if(Aborted)

Unlock any items that are locked and restart again with the same timestamp;

End Else;

End else

}

Step 5: End(commit) or Abort Transaction Method:

This method implements the logic to abort a transaction or commit a transaction.

If the input file reads 'e' then

abortTransaction method is called

Pseudocode:

abortTransaction(transactionId){

Item name, transaction id and transaction state are retrieved

If transaction state is "Active"

Set transaction to commit and release all the locks for items held by the transaction.

End if

Else

if transaction state is "Blocked"

Blocked function is called

End if

Else transaction state is “Abort”

Transaction is aborted

End else

End else}

Wound Wait method:

It compares the timestamp of the requested transaction with the one that has a lock on the data item.

Pseudocode:

woundWait(itemName,reqTransaction,heldTransaction, lock){

if reqTransaction.timeStamp less than heldTransaction.timeStamp

held transaction is aborted

if reqTransaction operation is read

set the lock state of the item to read and add transaction id to the item entry in lock table

end if

else reqTransaction operation is write

set the lock state of the item to write and add transaction id to the item entry in lock table

end else

end if

else

The request transaction state is said to block

end else

}

Execution Instructions:

This project is implemented using java. Main method is present in TwoPLProtocol.java

Compilation of java files:

Navigate to the code folder in command prompt

javac *.java

Execution:

java TwoPLProtocol

Enter the input file path when prompted

Note: The final input and output files are placed in “final input n output” folder.

Inputs and outputs for the project 1:

Input 1:

b1;
r1 (Y);
r1 (Z);
b2;
r2 (Y);
b3;
r3 (Y);
w1 (Z);
w3 (Y);
w2 (Y);
r2 (X);
e1;
e3;
w2 (X);
e2;

Output1:
Please enter the input transation file path:
C:\Users\PichikalaNagaVenkata\Desktop\DataBase2\final inputs\input1.txt

b1;
TID:1,operation:b,dataItem:null

Begin T1: Record is added to transaction table with Tid=1 and TS(T1)=1. T1 state=Active

r1(Y);
TID:1,operation:r,dataItem:Y
Y read locked by T1: Lock table record for Y is created with mode R (T1 holds lock).

r1(Z);
TID:1,operation:r,dataItem:Z
Z read locked by T1: Lock table record for Z is created with mode R (T1 holds lock).

b2;
TID:2,operation:b,dataItem:Z
Begin T2: Record is added to transaction table with Tid=2 and TS(T2)=2. T2 state=Active

r2(Y);
TID:2,operation:r,dataItem:Y
Y read locked by T2: Lock table record for Y is updated (T1, T2 hold R lock on Y)

b3;
TID:3,operation:b,dataItem:Y
Begin T3: Record is added to transaction table with Tid=3 and TS(T3)=3. T3 state=Active

r3(Y);
TID:3,operation:r,dataItem:Y
Y read locked by T3: Lock table record for Y is updated (T1, T2, T3 hold R lock on Y)

w1(Z);
TID:1,operation:w,dataItem:Z
Read lock upgraded to write lock for item Z by T1, lock table updated to mode W

w3(Y);
TID:3,operation:w,dataItem:Y
Read Write conflict between T3 and multiple Transactions in readlock list
T3 'BLOCKED'
Transaction T3 is blocked and this operation has been added to waiting list. T3 is waiting

w2(Y);
TID:2,operation:w,dataItem:Y
Read Write conflict between T2 and multiple Transactions in readlock list
T2 'BLOCKED'
Transaction T2 is blocked and this operation has been added to waiting list. T2 is waiting

r2(X);
TID:2,operation:r,dataItem:X
Transaction T2 is already BLOCKED
Entry for DataItem X has been made in the lock table
Transaction T2 is blocked and this operation has been added to waiting list. T2 is waiting

e1;
TID:1,operation:e,dataItem:X
Transaction T1 is committed and all the locks are released.
Transaction T3 is unblocked
Read Write conflict between T3 and multiple Transactions in readlock list
T3 'BLOCKED'
Transaction T3 is blocked and this operation has been added to waiting list. T3 is waiting
Operation was already blocked
Transaction T2 is unblocked
Read Write conflict between T2 and multiple Transactions in readlock list
Lock will be upgraded to 'Write Lock' from 'Read Lock' for data item Y on transaction id - 2
T3 is aborted and releases Read lock from item Y
Transaction T2 has been granted 'Write Lock' for data item Y
X read locked by T2: Lock table record for X is created with mode R (T2 holds lock).

e3;
TID:3,operation:e,dataItem:X
Transaction T3 is already aborted

w2(X);
TID:2,operation:w,dataItem:X
Read lock upgraded to write lock for item X by T2, lock table updated to mode W

e2;
TID:2,operation:e,dataItem:X
Transaction T2 is committed and all the locks are released.

Input 2:

b1;
r1(Y);
w1(Y);
r1(Z);
b2;
r2(Y);
b3;
r3(Z);
w1(Z);
e1;
w3(Z);
e3;
e2;

Output 2:

Please enter the input transation file path:

C:\Users\PichikalaNagaVenkata\Desktop\DataBase2\final inputs\input2.txt

b1;
TID:1,operation:b,dataItem:null
Begin T1: Record is added to transaction table with Tid=1 and TS(T1)=1. T1 state=Active

r1(Y);
TID:1,operation:r,dataItem:Y
Y read locked by T1: Lock table record for Y is created with mode R (T1 holds lock).

w1(Y);
TID:1,operation:w,dataItem:Y
Read lock upgraded to write lock for item Y by T1, lock table updated to mode W

r1(Z);
TID:1,operation:r,dataItem:Z
Z read locked by T1: Lock table record for Z is created with mode R (T1 holds lock).

b2;
TID:2,operation:b,dataItem:Z
Begin T2: Record is added to transaction table with Tid=2 and TS(T2)=2. T2 state=Active

r2(Y);
TID:2,operation:r,dataItem:Y
Write Read conflict between T2 and T1
T2 'BLOCKED'
Transaction T2 is blocked and this operation has been added to waiting list. T2 is waiting

b3;
TID:3,operation:b,dataItem:Y
Begin T3: Record is added to transaction table with Tid=3 and TS(T3)=3. T3 state=Active

r3(Z);
TID:3,operation:r,dataItem:Z
Z read locked by T3: Lock table record for Z is updated (T1, T3 hold R lock on Z)

w1(Z);
TID:1,operation:w,dataItem:Z
Read Write conflict between T1 and multiple Transactions in readlock list
Lock will be upgraded to 'Write Lock' from 'Read Lock' for data item Z on transaction id - 1
T3 is aborted and releases Read lock from item Z
Transaction T1 has been granted 'Write Lock' for data item Z

e1;
TID:1,operation:e,dataItem:Z
Transaction T1 is committed and all the locks are released.
Transaction T2 is unblocked

Transaction T2 has been appended to readlock list for data item Y. So, it has acquired 'Read Lock' on Y

w3(Z);
TID:3,operation:w,dataItem:Z
Transaction T3 is already aborted

e3;
TID:3,operation:e,dataItem:Z
Transaction T3 is already aborted

e2;
TID:2,operation:e,dataItem:Z
Transaction T2 is committed and all the locks are released.

Input 3:

b1;
r1 (Y);
r1 (Z);
b2;
r2 (Y);
b3;
r3 (Y);
w1 (Z);
e1;
w2 (Y);
r2 (X);
b4;
r4 (Z);
r4 (Y);
w2 (X);
e2;
w4 (Z);
e3;
w4 (Y);
e4;

Output 3:

Please enter the input transation file path:
C:\Users\PichikalaNagaVenkata\Desktop\DataBase2\final inputs\input3.txt

b1;
TID:1,operation:b,dataItem:null
Begin T1: Record is added to transaction table with Tid=1 and TS(T1)=1. T1 state=Active

r1(Y);
TID:1,operation:r,dataItem:Y
Y read locked by T1: Lock table record for Y is created with mode R (T1 holds lock).

r1(Z);
TID:1,operation:r,dataItem:Z
Z read locked by T1: Lock table record for Z is created with mode R (T1 holds lock).

b2;
TID:2,operation:b,dataItem:Z
Begin T2: Record is added to transaction table with Tid=2 and TS(T2)=2. T2 state=Active

r2(Y);
TID:2,operation:r,dataItem:Y
Y read locked by T2: Lock table record for Y is updated (T1, T2 hold R lock on Y)

b3;
TID:3,operation:b,dataItem:Y
Begin T3: Record is added to transaction table with Tid=3 and TS(T3)=3. T3 state=Active

r3(Y);
TID:3,operation:r,dataItem:Y
Y read locked by T3: Lock table record for Y is updated (T1, T2, T3 hold R lock on Y)

w1(Z);
TID:1,operation:w,dataItem:Z
Read lock upgraded to write lock for item Z by T1, lock table updated to mode W

e1;
TID:1,operation:e,dataItem:Z
Transaction T1 is committed and all the locks are released.

w2(Y);
TID:2,operation:w,dataItem:Y
Read Write conflict between T2 and multiple Transactions in readlock list
Lock will be upgraded to 'Write Lock' from 'Read Lock' for data item Y on transaction id - 2
T3 is aborted and releases Read lock from item Y
Transaction T2 has been granted 'Write Lock' for data item Y

r2(X);
TID:2,operation:r,dataItem:X
X read locked by T2: Lock table record for X is created with mode R (T2 holds lock).

b4;
TID:4,operation:b,dataItem:X
Begin T4: Record is added to transaction table with Tid=4 and TS(T4)=4. T4 state=Active

r4(Z);
TID:4,operation:r,dataItem:Z
Z read locked by T4: Lock table record for Z is created with mode R (T4 holds lock).

r4(Y);
TID:4,operation:r,dataItem:Y
Write Read conflict between T4 and T2
T4 'BLOCKED'
Transaction T4 is blocked and this operation has been added to waiting list. T4 is waiting

w2(X);
TID:2,operation:w,dataItem:X
Read lock upgraded to write lock for item X by T2, lock table updated to mode W

e2;
TID:2,operation:e,dataItem:X
Transaction T2 is committed and all the locks are released.
Transaction T4 is unblocked
Transaction T4 has been appended to readlock list for data item Y. So, it has acquired 'Read Lock' on Y

w4(Z);
TID:4,operation:w,dataItem:Z
Read lock upgraded to write lock for item Z by T4, lock table updated to mode W

e3;
TID:3,operation:e,dataItem:Z
Transaction T3 is already aborted

w4(Y);
TID:4,operation:w,dataItem:Y
Read lock upgraded to write lock for item Y by T4, lock table updated to mode W

e4;
TID:4,operation:e,dataItem:Y
Transaction T4 is committed and all the locks are released.

Input 4:

b1;
r1(Y);
w1(Y);
r1(Z);
b2;
r2(Y);
b3;
r3(Z);

w3(Z);
b4;
r4(X);
r4(Y);
e1;
w4(X);
e3;
e2;
w4(Y);
e4;

Output 4:

Please enter the input transation file path:
C:\Users\PichikalaNagaVenkata\Desktop\DataBase2\final inputs\input4.txt

b1;
TID:1,operation:b,dataItem:null
Begin T1: Record is added to transaction table with Tid=1 and TS(T1)=1. T1 state=Active

r1(Y);
TID:1,operation:r,dataItem:Y
Y read locked by T1: Lock table record for Y is created with mode R (T1 holds lock).

w1(Y);
TID:1,operation:w,dataItem:Y
Read lock upgraded to write lock for item Y by T1, lock table updated to mode W

r1(Z);
TID:1,operation:r,dataItem:Z
Z read locked by T1: Lock table record for Z is created with mode R (T1 holds lock).

b2;
TID:2,operation:b,dataItem:Z
Begin T2: Record is added to transaction table with Tid=2 and TS(T2)=2. T2 state=Active

r2(Y);
TID:2,operation:r,dataItem:Y
Write Read conflict between T2 and T1
T2 'BLOCKED'
Transaction T2 is blocked and this operation has been added to waiting list. T2 is waiting

b3;
TID:3,operation:b,dataItem:Y
Begin T3: Record is added to transaction table with Tid=3 and TS(T3)=3. T3 state=Active

r3(Z);

TID:3,operation:r,dataItem:Z
Z read locked by T3: Lock table record for Z is updated (T1, T3 hold R lock on Z)

w3(Z);
TID:3,operation:w,dataItem:Z
Read Write conflict between T3 and multiple Transactions in readlock list
T3 'BLOCKED'
Transaction T3 is blocked and this operation has been added to waiting list. T3 is waiting

b4;
TID:4,operation:b,dataItem:Z
Begin T4: Record is added to transaction table with Tid=4 and TS(T4)=4. T4 state=Active

r4(X);
TID:4,operation:r,dataItem:X
X read locked by T4: Lock table record for X is created with mode R (T4 holds lock).

r4(Y);
TID:4,operation:r,dataItem:Y
Write Read conflict between T4 and T1
T4 'BLOCKED'
Transaction T4 is blocked and this operation has been added to waiting list. T4 is waiting

e1;
TID:1,operation:e,dataItem:Y
Transaction T1 is committed and all the locks are released.
Transaction T2 is unblocked
Transaction T2 has been appended to readlock list for data item Y. So, it has acquired 'Read Lock' on Y
Transaction T3 is unblocked
Transaction T3 has been granted 'Write Lock' for data item Z

w4(X);
TID:4,operation:w,dataItem:X
Transaction T4 is already BLOCKED
Transaction T4 is blocked and this operation has been added to waiting list. T4 is waiting

e3;
TID:3,operation:e,dataItem:X
Transaction T3 is committed and all the locks are released.

e2;
TID:2,operation:e,dataItem:X
Transaction T2 is committed and all the locks are released.
Transaction T4 is unblocked
Y read locked by T4: Lock table record for Y is updated (T4 hold R lock on Y)
Read lock upgraded to write lock for item X by T4, lock table updated to mode W

w4(Y);
TID:4,operation:w,dataItem:Y
Read lock upgraded to write lock for item Y by T4, lock table updated to mode W

e4;
TID:4,operation:e,dataItem:Y
Transaction T4 is committed and all the locks are released.